**Java’s Compilation Process 2301921414 - Wais Ibrahim**

Java code consists of a two step execution process, first through an OS independent compiler and then through the Java Virtual Machine (JVM) which is built unique to each operating system. These steps consist of the following:

1. **Compilation**

Firstly, the compiler that is a native executable called ‘javac.exe’ takes the .java file and transforms the source code into a machine independent encoding, called Bytecode. While source code is converted into bytecode, the compiler executes the following:

* **Parse**: A set of \*.java source files are read and mapped into AST (Abstract Syntax Tree)-Nodes.
* **Enter**: Enters symbols for the definitions into the symbol table.
* **Process annotations**: If requested, processes annotations found in the specified compilation units.
* **Attribute**: Syntax trees are then attributed to error check. This includes name resolution, type checking and constant folding.
* **Flow**: Dataflow analysis is performed on trees stated previously, including checks for assignment and reachability.
* **Desugar**: Rewrites AST and translates away syntactic sugar.
* **Generate**: Generates ‘.Class’ files.

1. **Execution**

This step will execute the compiled code. The bytecode is passed to the JVM, where Java code is executed to produce an output. Java provides platform specific JVMs that allow our Java code to be run on several different platforms such as Mac, and Windows, among others; this means that Windows as an example, has its specific JVM that executes Java code. Next, the main class file is passed to the JVM, and procedurally goes through three main stages before execution.

* Class Loader: the main class is loaded into memory. All other classes in the program are loaded through the class loader.
* Bytecode Verifier: The Bytecode Verifier will check each instruction to make sure that they aren’t damaging such as checking if variables are initialised before they’re used, checking rules for access modifiers are correctly used.

Finally, the loaded bytecode must be converted into machine code. Using a Just-In-Time compiler (JIT), the hardware executes native code.